End-to-end Automatic Speech Recognition

Markus Nussbaum-Thom

IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598, USA
Markus Nussbaum-Thom.

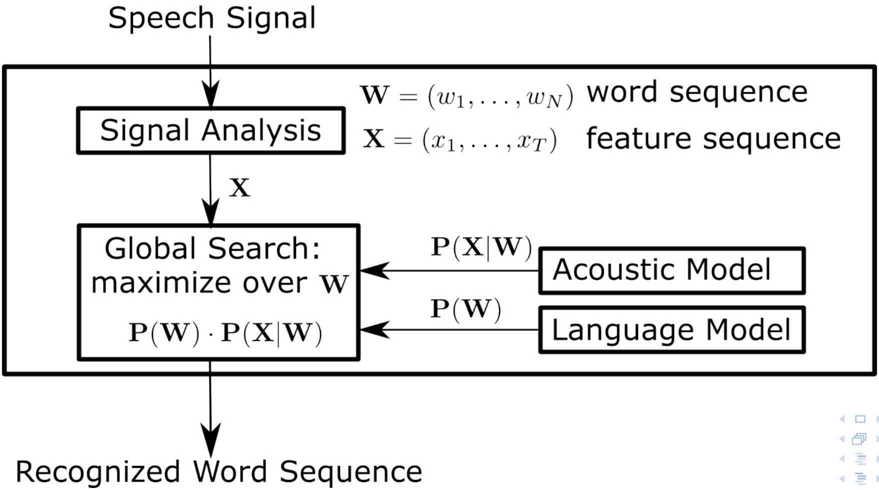
February 22, 2017

Contents

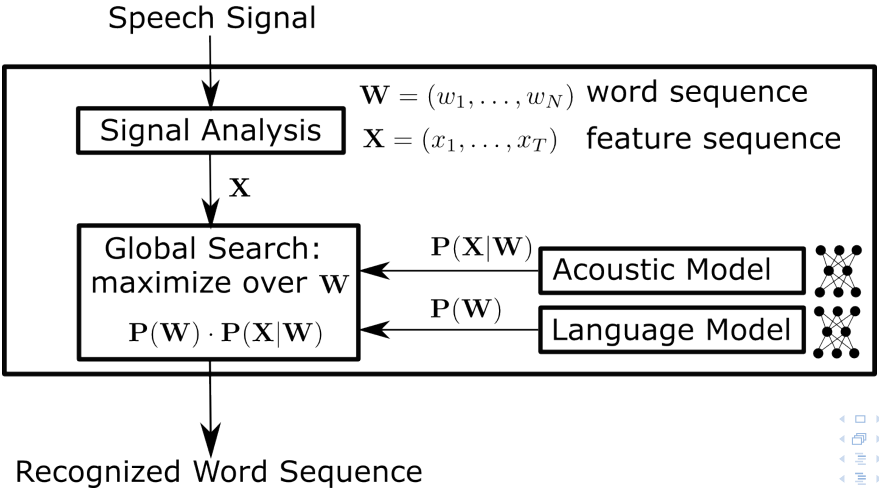
Terminology

- Features: $x, x_t, x_1^T := x_1, \ldots, x_T$.
- Words: $w, u, v, w_m, w_1^M := w_1, \ldots, w_M$.
- Word sequences: W, W_n, V.
- States: $s, s_t, s_1^T := s_1, \ldots, s_T$.
- Class conditional posterior probability: $p(s_t|x_t)$, $p(W, s_1^T|x_1^T)$.

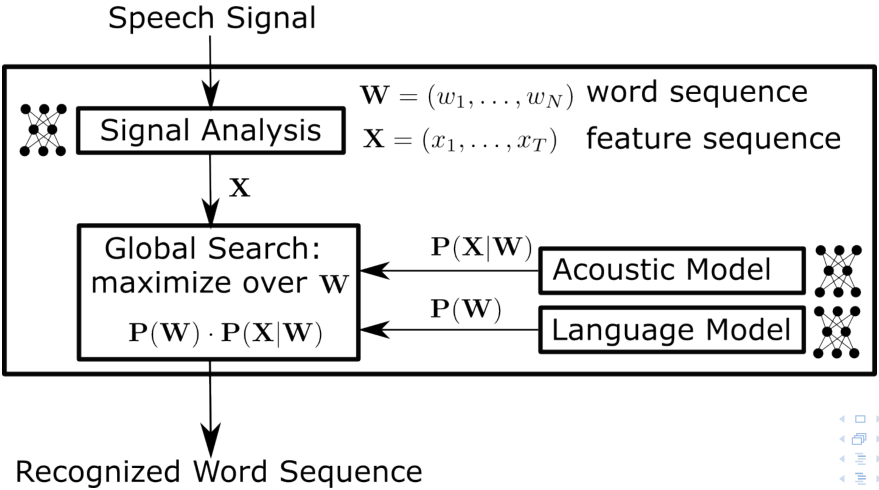
Bayes' Decision Rule



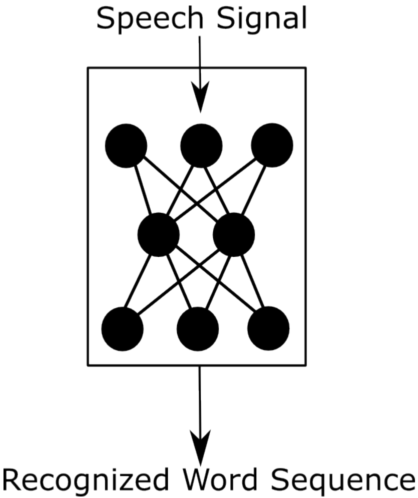
Towards End-to-End Automatic Speech Recognition



Towards End-to-End Automatic Speech Recognition



End-to-End Automatic Speech Recognition



End-to-End Approach

- End-to-end:
  - Training all modules to optimize a global performance criterion. (LeCun et al., 98)
- Easy: Classes do not have a sub-structure
  - e.g. image classification.
- Difficult: Classes have a sub-structure (sequences, graphs)
  - e.g. automatic speech recognition,
  - automatic handwriting recognition,
  - machine translation.
- Segmentation problem: Which part of the input related to which part of the sub-structure ?

Towards End-to-end Automatic Speech Recognition

- End-to-end acoustic model:
  - Using characters instead of phonemes.
  - Connectionst temporal classification using recurrent or convolutional neural networks.
  - Purely neural attention model.
- End-to-end feature extraction:
  - Feature extraction integrated into the acoustic model.
  - Using the raw time signal.
  - Learning a specific type of filter.
- Towards real end-to-end modeling:
  - Using word as targets instead of characters or phonemes and a massive amount of data.

Basic Problem

- Input: $X = x_1^T = (x_1, \ldots, x_T)$
- Neural network: $p(\cdot|x_1), \ldots, p(\cdot|x_T)$.
- Target: $W = w_1^M = (w_1, \ldots, w_M)$
- but $M << T$
- How do we solve this ?
- Connectionist Temporal Classification (CTC).
  [Graves et al., 2006, Graves et al., 2009, CTC]
- Attention Models.
  [Bahdanau et al., 2016, Chorowski et al., 2015, Chorowski et al., 2015, Attention]
- Inverted Hidden Markov Models.
  [Doetsch et al., 2016, Inverted HMM - a Proof of Concept]

Overview CTC

- Concept.
- Training.
- Recognition.

Connectionist Temporal Classification (CTC)

- Given X = (x_1, . . . , x_5) and W = (a, b, c)
- Introduce blank state and allow word repititions: $\square$

x_1	x_2	x_3	x_4	x_5
a	b	c	$\square$	$\square$
a	$\square$	b	c	$\square$
a	a	b	c	$\square$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\square$	$\square$	a	b	c

- Blank and repitition removal $\mathcal{B}$: $\mathcal{B}(a, \square, b, c, \square, \square) = (a, b, c)$

Connectionist Temporal Classification (CTC)

- Posterior for sentence $W = w_1^M$ and features $X = x_1^T$:

$$p(W|X) = \sum_{s_1^T \in \mathcal{B}^{-1}(W)} p(s_1^T|X)$$

$$:= \sum_{s_1^T \in \mathcal{B}^{-1}(W)} \prod_{t=1}^{T} p(s_t|x_t)$$

- Training criterion for training samples $(X_n, W_n), n = 1, \ldots, N$:

$$\mathcal{F}_{\mathrm{CTC}}(\Lambda) = -\frac{1}{N} \sum_{n=1}^{N} \log p_\Lambda(W_n|X_n)$$

Overview CTC

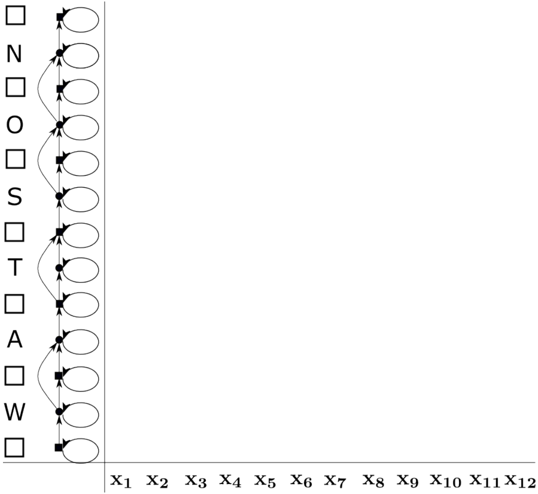
- Concept.
- Training.
- Recognition.

Forward-Backward Decomposition (CTC)

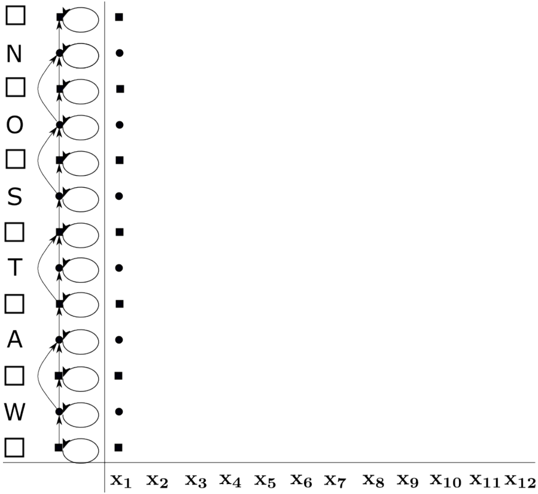
- $\alpha(t,m,v)$: Sum over $s_1^t \in B(w_1^m)$ for given x_1^t ending in v.
- $\beta(t,m,v)$: Sum over $s_t^T \in B(w_m^M)$ for given x_t^T starting in v.
- Choose $t \in 1,\dots,T$:

$$
\begin{aligned}
p(w_1^M|x_1^T) &= \sum_{s_1^T \in \mathcal{B}^{-1}(w_1^M)} p(s_1^T|x_1^T) \\
&= \dots \\
&= \sum_{m=1}^{M} \sum_{v \in \{w_m, \square\}} \frac{\alpha(t,m,v)}{p(v|x_t)} \cdot p(v|x_t) \cdot \frac{\beta(t,m,v)}{p(v|x_t)}
\end{aligned}
$$

Forward Algorithm (CTC)

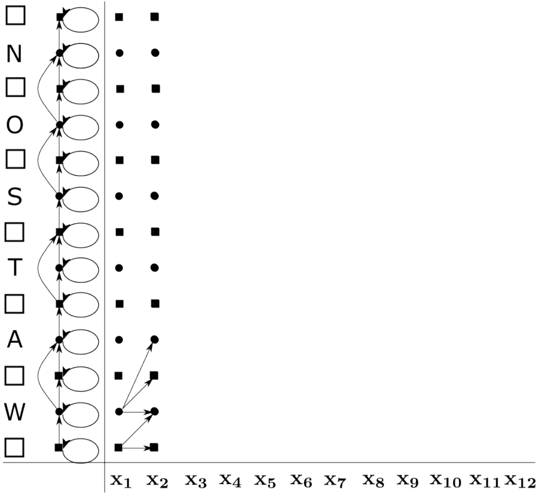


Forward Algorithm (CTC)



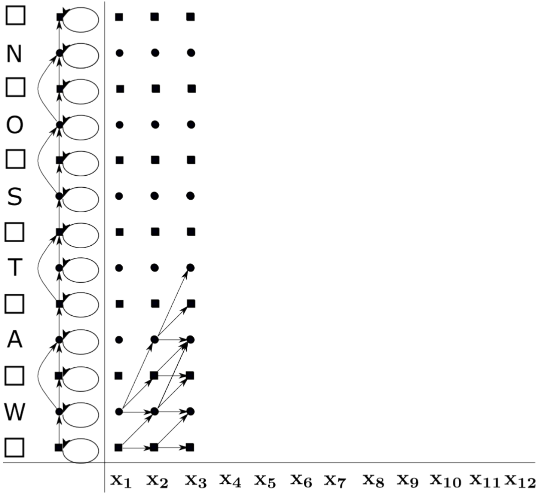
- Compute $\alpha(1, m, v) = p(v|x_1)$

Forward Algorithm (CTC)



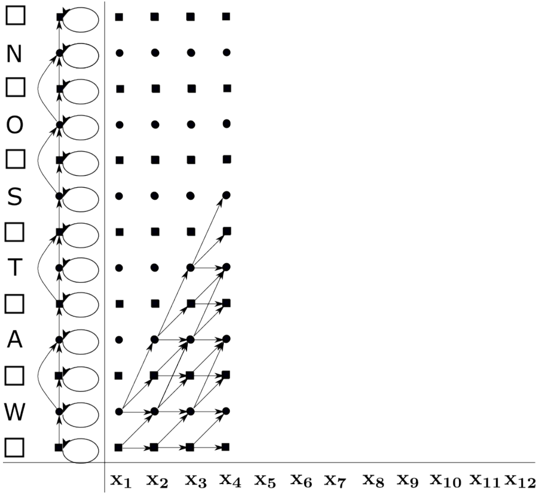
- Compute $\alpha(2, m, v)$.

Forward Algorithm (CTC)



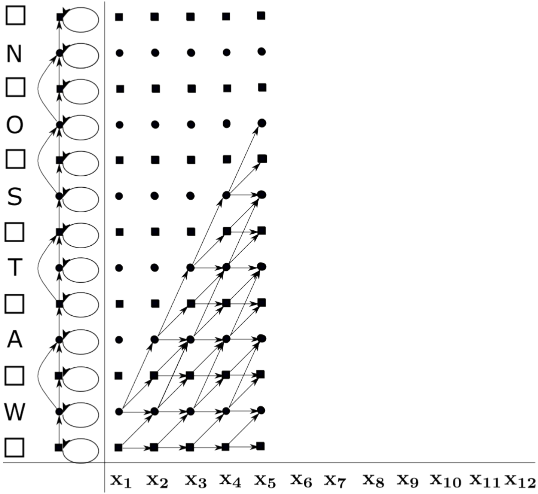
- Compute $\alpha(3, m, v)$.

Forward Algorithm (CTC)



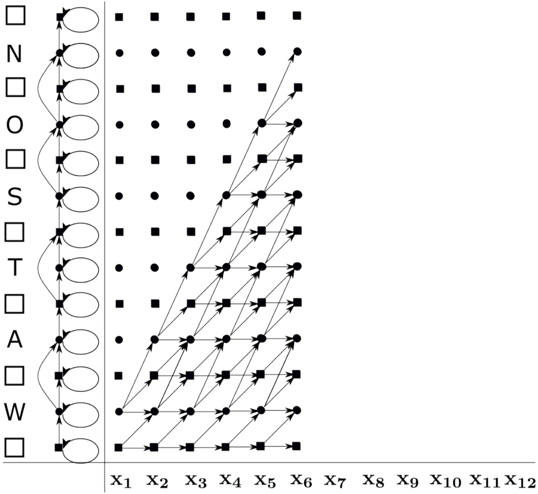
- Compute $\alpha(4, m, v)$.

Forward Algorithm (CTC)



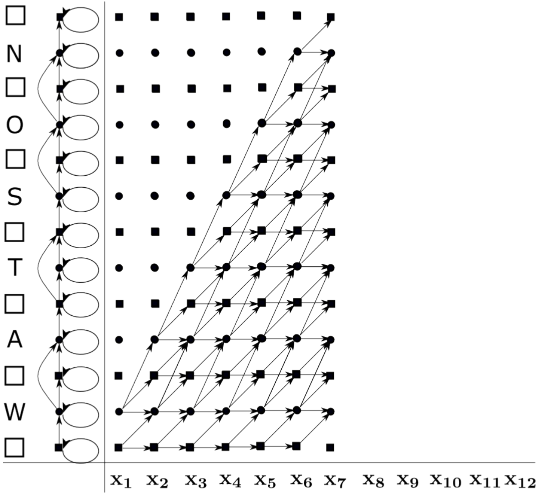
- Compute $\alpha(5, m, v)$.

Forward Algorithm (CTC)



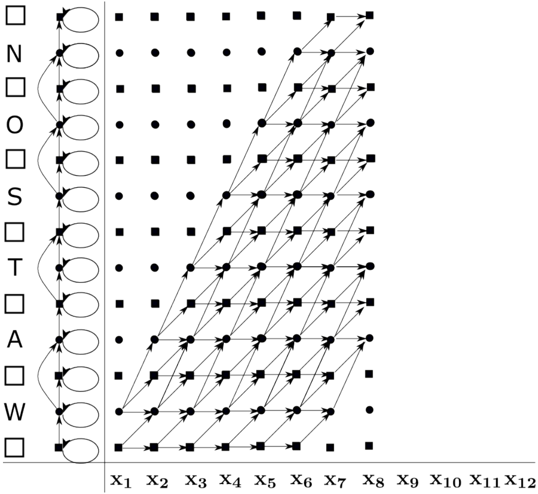
- Compute $\alpha(6, m, v)$.

Forward Algorithm (CTC)



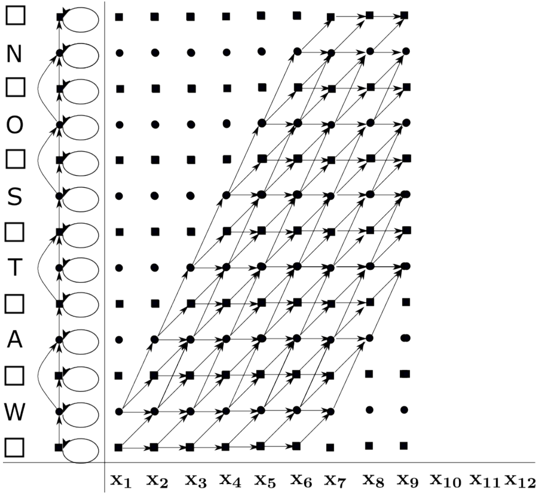
- Compute $\alpha(7, m, v)$.

Forward Algorithm (CTC)



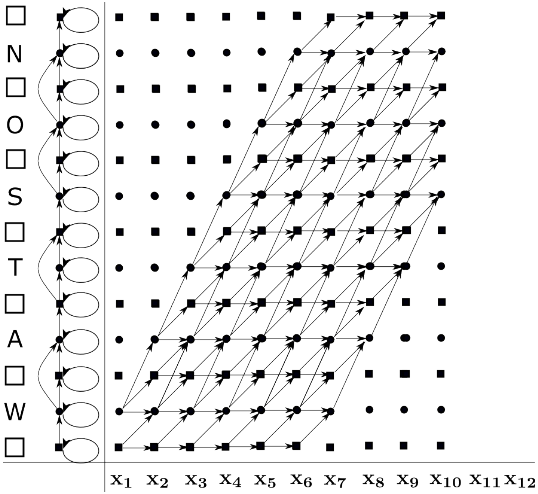
- Compute $\alpha(8, m, v)$.

Forward Algorithm (CTC)



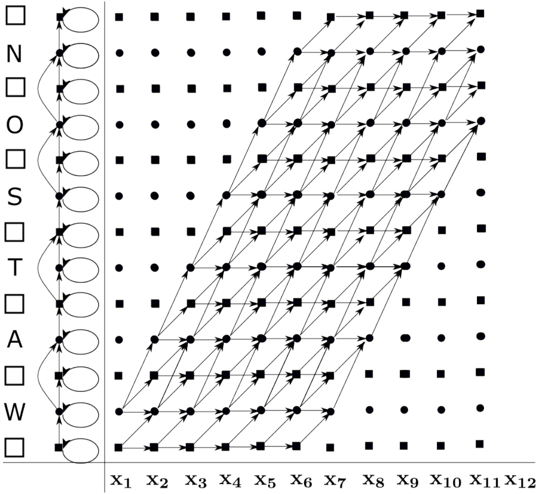
- Compute $\alpha(9, m, v)$.

Forward Algorithm (CTC)



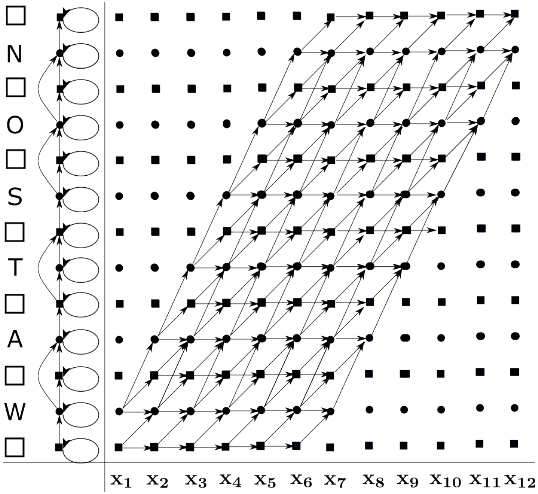
- Compute $\alpha(10, m, v)$.

Forward Algorithm (CTC)



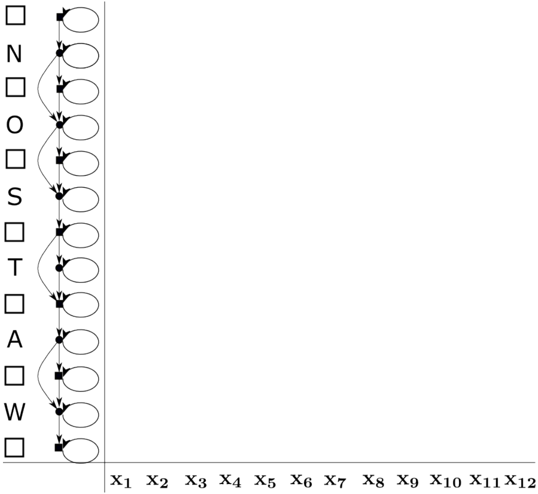
- Compute $\alpha(11, m, v)$.

Forward Algorithm (CTC)

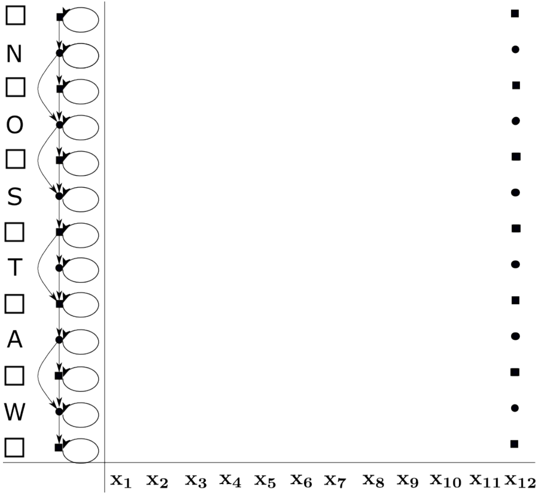


- Compute $\alpha(12, m, v)$.

Backward Algorithm (CTC)

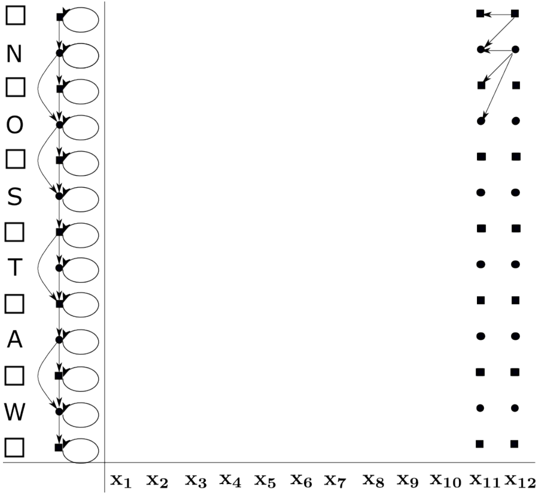


Backward Algorithm (CTC)



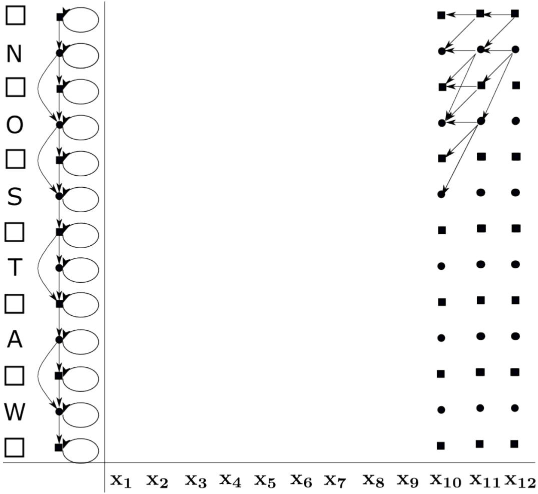
- Compute $\beta(12, M, v) = p(v|x_1 2)$

Backward Algorithm (CTC)



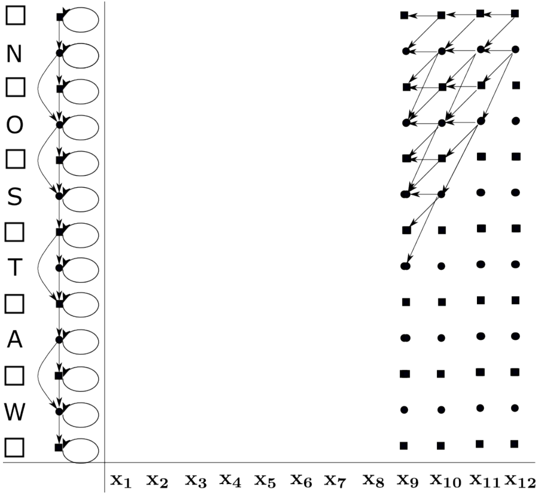
- Compute $\beta(11, m, v)$.

Backward Algorithm (CTC)



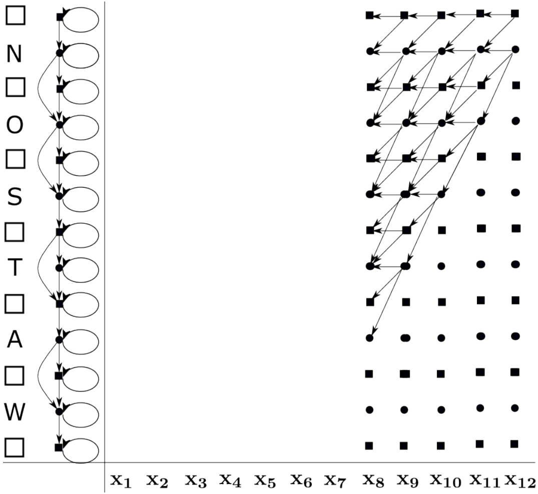
- Compute $\beta(10, m, v)$.

Backward Algorithm (CTC)



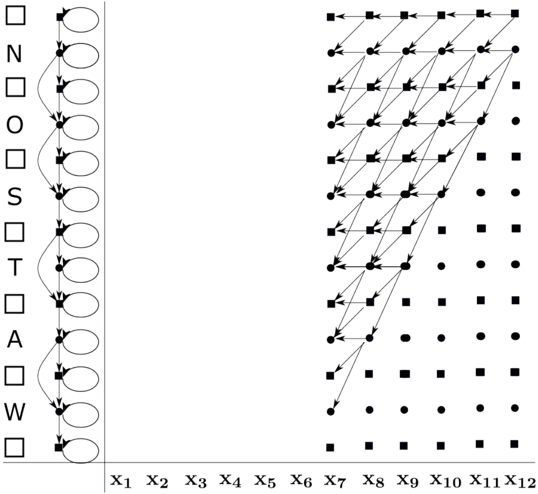
- Compute $\beta(9, m, v)$.

Backward Algorithm (CTC)



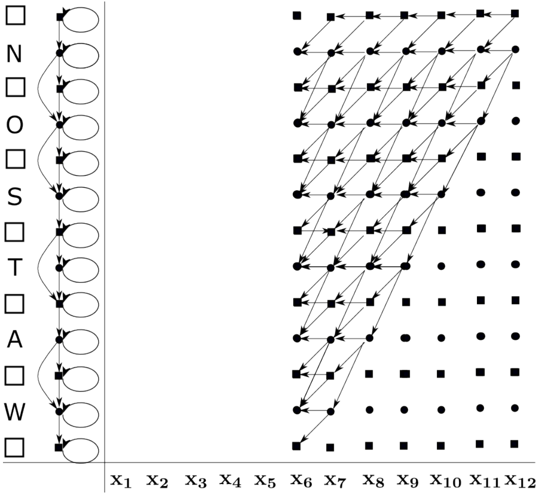
- Compute $\beta(8, m, v)$.

Backward Algorithm (CTC)



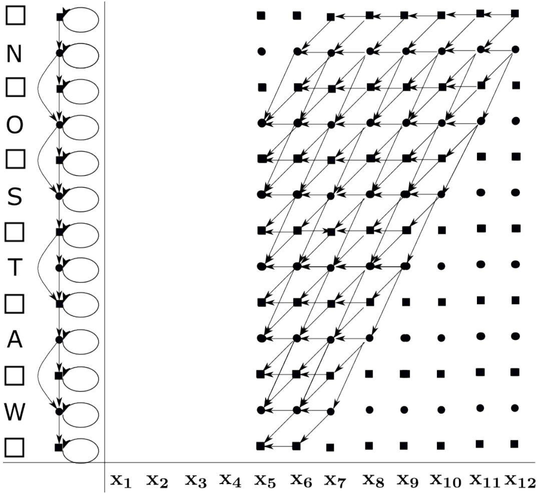
- Compute $\beta(7, m, v)$.

Backward Algorithm (CTC)



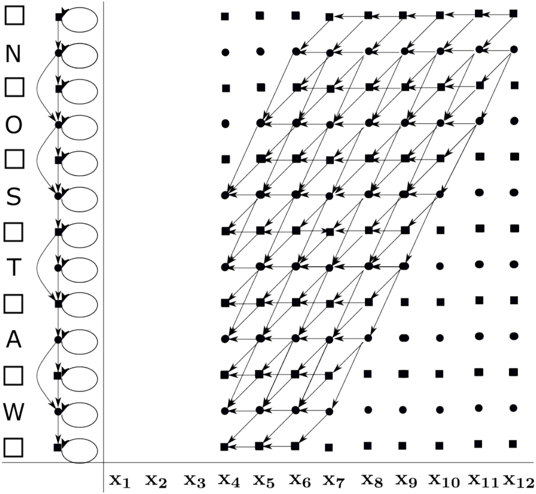
- Compute $\beta(6, m, v)$.

Backward Algorithm (CTC)



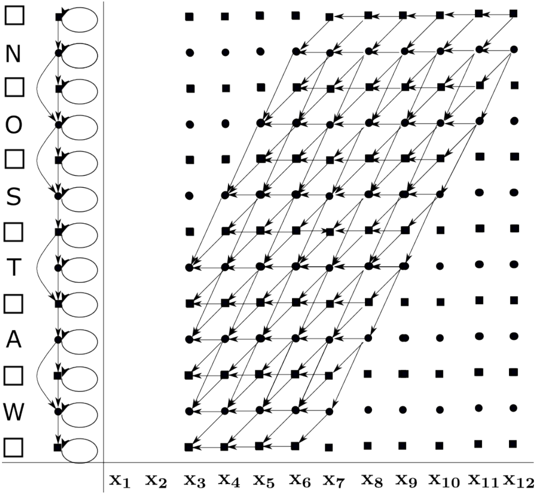
- Compute $\beta(5, m, v)$.

Backward Algorithm (CTC)



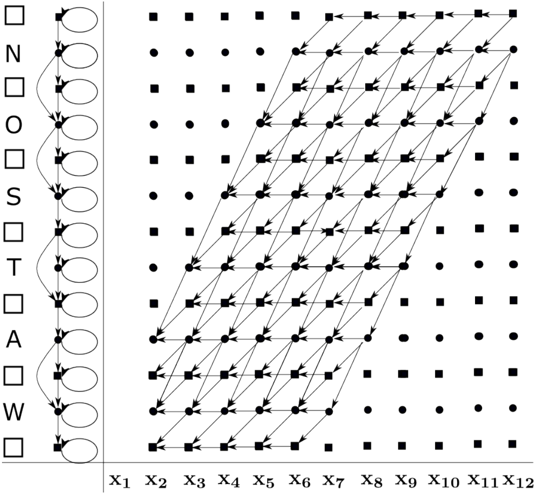
- Compute $\beta(4, m, v)$.

Backward Algorithm (CTC)



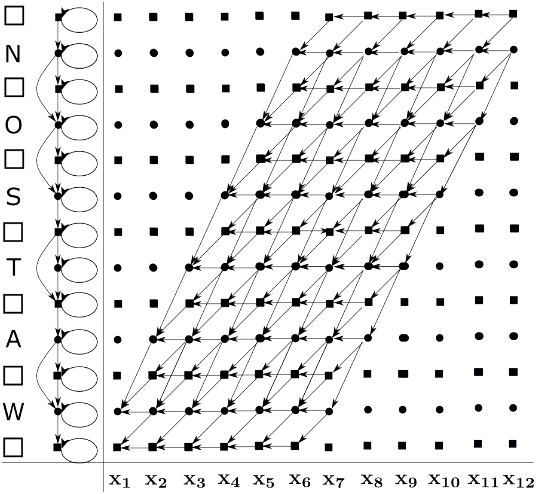
- Compute $\beta(3, m, v)$.

Backward Algorithm (CTC)



- Compute $\beta(2, m, v)$.

Backward Algorithm (CTC)



- Compute $\beta(1, m, v)$.

Posterior Decomposition (CTC)

Choose $t \in 1, \ldots, T$:

$$p(w_1^M|X) = \sum_{s_1^T \in \mathcal{B}^{-1}(w_1^M)} p(s_1^T|X) \qquad \text{"definition"}$$

$$= \sum_{s_1^T \in \mathcal{B}^{-1}(w_1^M)} \prod_{\tau=1}^{T} p(s_\tau|x_\tau) \qquad \text{"model assumption"}$$

$$= \sum_{m=1}^{M} \sum_{v \in \{w_m, \square\}} \sum_{\substack{s_1^T \in \mathcal{B}^{-1}(w_1^M) \\ s_t = v}} \prod_{\tau=1}^{T} p(s_\tau|x_\tau)$$

$$\text{"decomposition around } t\text{"}$$

$$= \sum_{m=1}^{M} \sum_{v \in \{w_m, \square\}} \sum_{\substack{s_1^T \in \mathcal{B}^{-1}(w_1^M) \\ s_t = v}} \prod_{\tau=1}^{t-1} p(s_\tau|x_\tau) \cdot p(s_v|x_t) \cdot \prod_{\rho=t+1}^{T} p(s_\rho|x_\rho)$$

Posterior Decomposition (CTC)

$$= \sum_{m=1}^{M} \sum_{v \in \{w_m, \Box\}} \sum_{\substack{s_1^T \in \mathcal{B}^{-1}(w_1^M) \\ s_t = v}} \prod_{\tau=1}^{t-1} p(s_\tau | x_\tau) \cdot p(s_v | x_t) \cdot \prod_{\rho=t+1}^{T} p(s_\rho | x_\rho)$$

$$= \sum_{m=1}^{M} \sum_{v \in \{w_m, \Box\}} \sum_{\substack{s_1^T \in \mathcal{B}^{-1}(w_1^M) \\ s_t = v}} \frac{\prod_{\tau=1}^{t} p(s_\tau | x_\tau)}{p(v | x_t)} \cdot p(v | x_t) \cdot \frac{\prod_{\rho=t}^{T} p(s_\rho | x_\rho)}{p(v | x_t)}$$

Posterior Decomposition (CTC)

$$= \sum_{m=1}^{M} \sum_{v \in \{w_m, \square\}} \frac{\sum\limits_{\substack{s_1^t \in \mathcal{B}^{-1}(w_1^m) \\ s_t = v}} \prod\limits_{\tau=1}^{t} p(v|x_\tau)}{p(v|x_t)} \cdot p(v|x_t) \cdot \frac{\sum\limits_{\substack{s_t^T \in \mathcal{B}^{-1}(w_m^M) \\ s_t = v}} \prod\limits_{\rho=t}^{T} p(s_\rho|x_\rho)}{p(v|x_t)}$$

$$= \sum_{m=1}^{M} \sum_{v \in \{w_m, \square\}} \frac{\alpha(t,m,v)}{p(v|x_t)} \cdot p(v|x_t) \cdot \frac{\beta(t,m,v)}{p(v|x_t)}$$

- $\alpha(t,m,v)$: Sum over $s_1^t \in B(w_1^m)$ for given x_1^t ending in v.
- $\beta(t,m,v)$: Sum over $s_t^T \in B(w_m^M)$ for given x_t^T starting in v.

Forward Path Decomposition (CTC)

- Consider a path $s_1^t \in \mathcal{B}^{-1}(w_1^m), s_t = v$:
  - $s_t = w_m$

s_1^{t-1}	s_{t-1}	s_t
$w_1 \dots w_?$	?	w_m
$w_1 \dots w_m$	w_m	w_m
$w_1 \dots w_{m-1}$	$\square$	w_m
$w_1 \dots w_{m-1}$	w_{m-1}	w_m

  - $s_t = \square$

s_1^{t-1}	s_{t-1}	s_t
$w_1 \dots w_?$	?	$\square$
$w_1 \dots w_m$	w_m	$\square$
$w_1 \dots w_m$	$\square$	w_m

Forward Probablities (CTC)

$$\alpha(t,m,v) = \sum_{\substack{s_1^t \in \mathcal{B}^{-1}(w_1^m) \\ w_m = v}} \prod_{\tau=1}^{t} p(s_\tau|x_\tau)$$

$$= p(v|x_t) \cdot \begin{cases} \displaystyle\sum_{u \in \{w_{m-1}, \square\}} \sum_{\substack{s_1^{t-1} \in \mathcal{B}^{-1}(w_1^{m-1}) \\ s_{t-1} = u}} \prod_{\tau=1}^{t-1} p(s_\tau|x_\tau) \\ \displaystyle + \sum_{\substack{s_1^{t-1} \in \mathcal{B}^{-1}(w_1^m) \\ s_{t-1} = w_m}} \prod_{\tau=1}^{t-1} p(s_\tau|x_\tau), & v = w_m \\ \displaystyle\sum_{u \in \{w_m, \square\}} \sum_{\substack{s_1^{t-1} \in \mathcal{B}^{-1}(w_1^m) \\ s_{t-1} = u}} \prod_{\tau=1}^{t-1} p(s_\tau|x_\tau), & v = \square \end{cases}$$

Forward Probabilities (CTC)

$$= p(v|x_t) \cdot \begin{cases} \sum\limits_{u \in \{w_{m-1}, \square\}} \alpha(t-1, m-1, u) + \alpha(t-1, m, w_m) & , v = w_m \\ \sum\limits_{u \in \{w_m, \square\}} \alpha(t-1, m, u) & , v = \square \end{cases}$$

Backward Probablities (CTC)

$\beta(t, m, v) =$ Sum over all pathes $s_t^T \in B(w_m^M)$ for given x_t^T
starting in a word v.

$$\beta(t, m, v) = \sum_{\substack{s_t^T \in \mathcal{B}^{-1}(w_m^M) \\ w_m = v}} \prod_{\tau=t}^{T} p(s_\tau | x_\tau)$$

$$= \begin{cases} p(v|x_t) \cdot \sum_{u \in \{w_{m+1}, \Box\}} \beta(t+1, m+1, u) + \beta(t+1, m, w_m), & v = w_m \\ p(\Box|x_t) \cdot \sum_{u \in \{w_m, \Box\}} \beta(t+1, m, u) & , v = \Box \end{cases}$$

Derivatives CTC

- Derivative posterior:

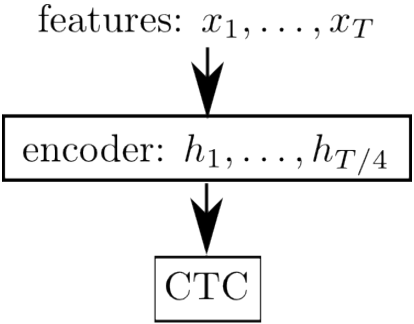
$$
\begin{aligned}
&\nabla_{p(s|x_t)} P(W|X) \\
&= \nabla_{p(s|x_t)} \sum_{m=1}^{M} \sum_{v \in \{w_m, \square\}} \frac{\alpha(t,m,v)}{p(v|x_t)} \cdot p(v|x_t) \cdot \frac{\beta(t,m,v)}{p(v|x_t)} \\
&= \sum_{m=1}^{M} \sum_{v \in \{w_m, \square\}} \delta(v,s) \frac{\alpha(t,m,s) \cdot \beta(t,m,s)}{p^2(s|x_t)}
\end{aligned}
$$

$$
\nabla \log P(W|X) = \frac{1}{P(W|X)} \nabla P(W|X)
$$

- Derivative training criterion:

$$
\Rightarrow \nabla \mathcal{F}_{\mathrm{CTC}}(\Lambda) = -\frac{1}{N} \sum_{n=1}^{N} \nabla \log p(W_n|X_n)
$$

CTC Architectures

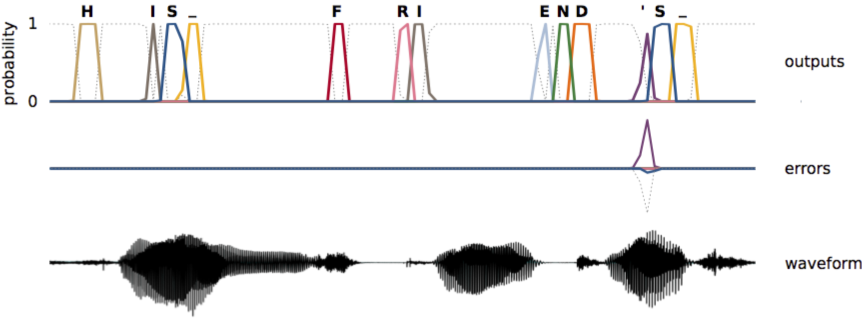


- What kind of encoders ? DNNs, (bidirectional) LSTMs, CNNs.
- Subsampling: Reducing framerate through the network.

Subsampling

- Join input frames.
- Reshape input to next layer:
- Return every 2nd frame to the next layer.
- CNNs: Use strides.

Peaking Behavior



[Graves and Jaitly, 2014, citation]

Overview CTC

- Concept.
- Training.
- Recognition.

Hybrid Recognition (CTC)

- Hybrid:
  - Model:

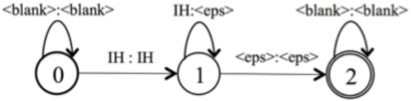
$$p(x|s) \sim \frac{p(s|x)}{p(s)} = \frac{p(s|x)}{p(x)p(s)} =$$

  - Decoding:

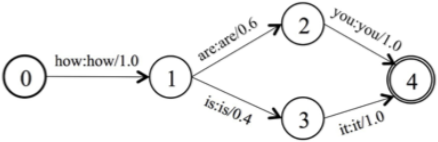
$$\hat{w}_1^N = \arg\max_{w_1^N} \left\{ p(w_1^N) \max_{s_1^T} \prod_{\tau=1}^{T} p(x_t|s_t) \right\}$$

Weighted Finite State Transudcer Recognition (WFST)

- Token T:



- Language Model G:



- Lexicon L:

- Search Space:

$$S = T \circ \min(\det(L \circ G))$$

Resources for CTC

- Keras:
  - Tensorflow: https://github.com/fchollet/keras/blob/master/keras/backend/tensorflow_backend.py
  - Theano: https://github.com/fchollet/keras/blob/master/keras/backend/theano_backend.py
  - Example: https://github.com/fchollet/keras/blob/master/examples/image_ocr.py
- Baidu:
  - https://github.com/baidu-research/warp-ctc
  - https://github.com/sherjilozair/ctc
  - https://github.com/baidu-research/ba-dls-deepspeech
- Eesen: https://github.com/srvk/eesen
- Kaldi: https://github.com/lingochamp/kaldi-ctc

Further Literature on CTC

[Miao et al., 2015, EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding]

[Collobert et al., 2016, Wav2Letter: an End-to-End ConvNet-based Speech Recognition System]

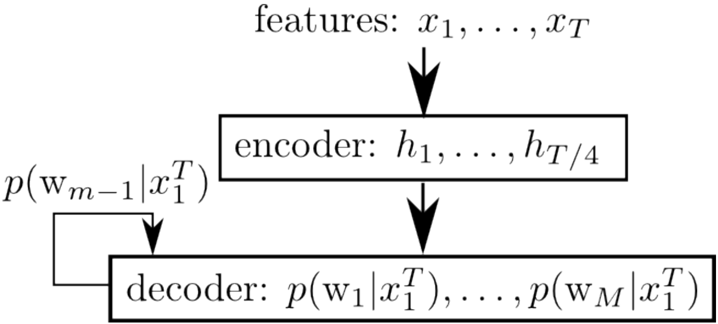
[Zhang et al., 2017, Towards End-to-End Speech Recognition with Deep Convolutional Neural Networks]

[Senior et al., 2015, Acoustic modelling with CD-CTC-SMBR LSTM RNNS]

[Soltau et al., 2016, Neural Speech Recognizer: Acoustic-to-Word LSTM Model for Large Vocabulary Speech Recognition]

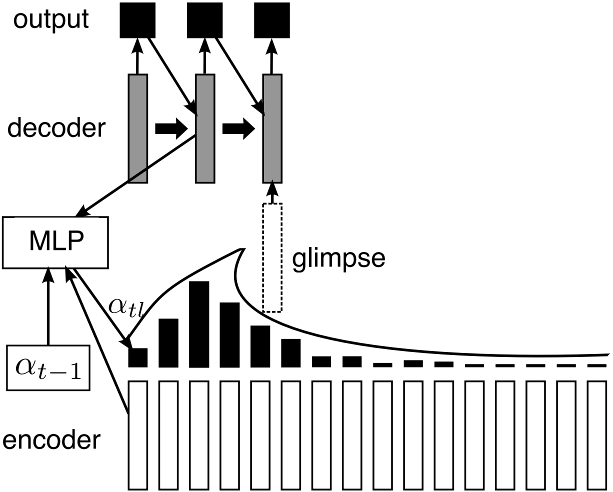
Attention Model

- Encoder-Decoder architecture:
  - Encoder: performs a feature extracation/encoding based on the input.
  - Decoder: Produces output sequence output labels from the encoded features.



Attention Encoder-Decoder Architecture

- What kind of encoders ? DNNs, (bidirectional) LSTMs, CNNs.



Attention Encoder-Decoder Architecture

- Encoder-Decoder:
  - Input: $x_1^T = x_1, \ldots, x_T$
  - Encoder: $h_1^{T/4} = h_1, \ldots, h_{T/4} = \mathrm{Encoder}(x_1^T)$
  - Decoder: For $m = 1, \ldots, M$:
    - Attention: $\alpha_m = \mathrm{Attend}(h_1^{T/4}, s_{m-1}, \alpha_{m-1})$
    - Glimpse: $g_m = \sum_{\tau=1}^{T/4} \alpha_{m,t} h_t$
    - Generator: $y_m = \mathrm{Generator}(g_m, s_{m-1})$

      $$c_m = \mathrm{RNN}(c_{m-1}, g_m, s_{m-1})$$

      $$y_m = \mathrm{Softmax}(c_m)$$
    - Transition: $s_m = \mathrm{RNN}(s_{m-1}, y_m, g_m)$

Attention Mechanism

- Content based: (weights: E, W, V and bias: b)

$$\epsilon_{m,t} = E \cdot \tanh(W \cdot s_{m-1} + V \cdot h_t + b)$$

- Location based: (weights: E, W, V, U and bias: b)

$$\begin{aligned} f &= F * \alpha_{m-1} \\ \epsilon_{m,t} &= E \cdot \tanh(W \cdot s_{m-1} + V \cdot h_t + U \cdot f_{m,t} + b) \end{aligned}$$

- Renormalization: (sharpening: γ)

$$\alpha_{m,t} = \frac{\exp(\gamma \cdot \epsilon_{m,t})}{\sum_{t=1}^{T/4} \exp(\gamma \cdot \epsilon_{m,t})}$$

Window Around Median

- Compute median:

$$\tau_m = \arg \min_{k=1,\ldots,T/4} \left| \sum_{\rho=1}^{k} \alpha_{m-1,\rho} - \sum_{\theta=k+1}^{k} \alpha_{m-1,\theta} \right|$$

- Compute attention around median:

$$T_m = \{\tau_m - \omega_{\mathrm{left}}, \ldots, \tau_m + \omega_{\mathrm{right}}\}$$

$$\alpha_{m,t} = \begin{cases} \dfrac{\exp(\gamma \cdot \epsilon_{m,t})}{\sum\limits_{\tau \in T_m} \exp(\gamma \cdot \epsilon_{m,\tau})} & , t \in T_m \\ 0 & , \text{otherwise} \end{cases}$$

Other Techniques (Attention)

- Monotonic regularization:

$$r_m = \max\left\{0, \sum_{\tau=1}^{T/4}\left(\sum_{i=1}^{\tau}\alpha_{m,i} - \sum_{i=1}^{\tau}\alpha_{m-1,i}\right)\right\}$$

- Curriculum learning: Starting with shorter sequences and gradually increase sequence length.

- Flatstart: Initial positions are chosen according to speaker speed.

Resources for Attention

- Theano+Bricks+Blocks:
  `https://github.com/rizar/attention-lvcsr`
- Tensorflow:
  `https://www.tensorflow.org/tutorials/seq2seq`
- Keras: `https://github.com/farizrahman4u/seq2seq`

Further Literature on Attention

[Bahdanau et al., 2016, End-to-end attention-based large vocabulary speech recognition]

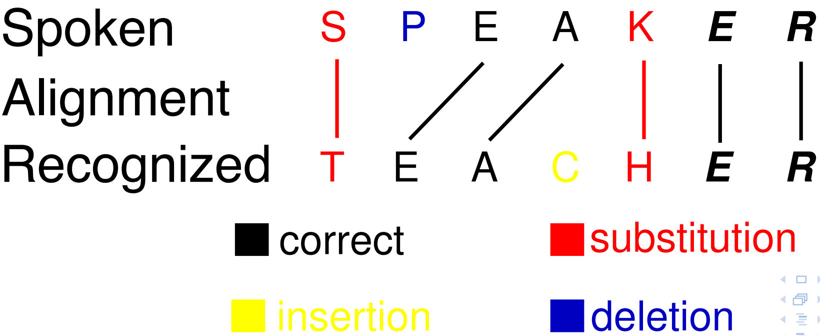
[Chorowski et al., 2015, Attention-Based Models for Speech Recognition]

[Chorowski et al., 2015, End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results]

[Kim et al., 2016, Joint CTC-Attention based End-to-End Speech Recognition using Multi-task Learning]

Evaluation Framework

- Development and evaluation set different from training set.
- Levenshtein: Minimum insertions, deletions and substitutions



Evaluation Framework

- Levenshtein: Minimum insertions, deletions and substitutions

$$L(w_1^N, v_1^M) = \min_{s,t} \left\{ \sum_{i=1}^{\lambda} \left(1 - \delta(w_{s(i)}, v_{t(i)})\right) \right\}$$

with dem Kronecker delta $\delta(w, v) = \begin{cases} 1 & , v = w \\ 0 & , v \neq w \end{cases}$

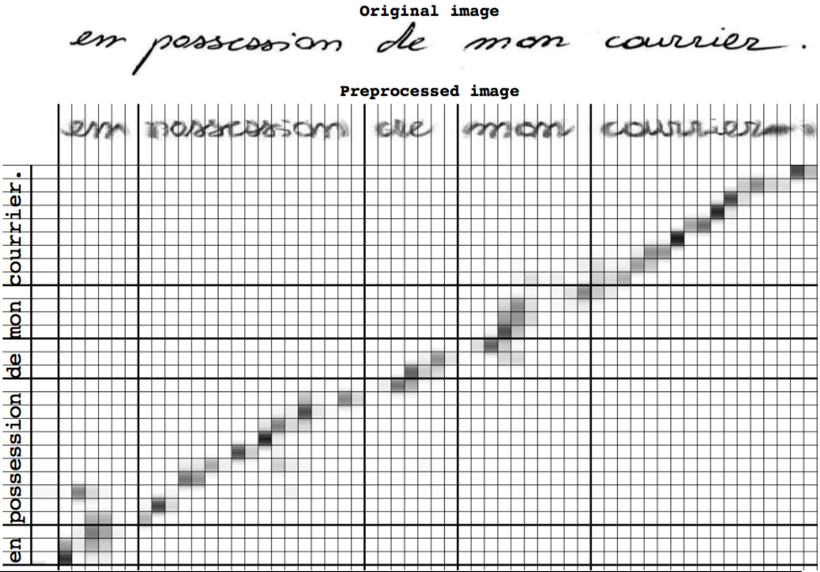
- Word Error Rate (WER):

$$WER(\text{Spoken}_1^R, \text{Recognized}_1^R) = \frac{\sum_{r=1}^{R} L(\text{Spoken}_r, \text{Recognized}_r)}{\sum_{r=1}^{R} |\text{Spoken}_r|}$$

Experimental Results

Model	CER	WER
Bhadanau et al. (2015)		
Attention	6.4	18.6
Attention + bigram LM	5.3	11.7
Attention + trigram LM	4.8	10.8
Attention + extended trigram LM	3.9	9.3
Graves and Jaitly (2014)		
CTC	9.2	30.1
Hannun et al. (2014)		
CTC + bigram LM	n/a	14.1
Miao et al. (2015)		
CTC + bigram LM	n/a	26.9
CTC for phonemes + lexicon	n/a	26.9
CTC for phonemes + trigram LM	n/a	7.3
CTC + trigram LM	n/a	9.0
Hybrid BGRU (15 h)	n/a	2.0

Attention Modeling Example from Handwriting



Inverted Hidden Markov Model (HMM)

- Traditional HMM:

$$
\begin{aligned}
p(w_1^N, x_1^T) &= p(w_1^N) \cdot p(x_1^T | w_1^N) \\
&= p(w_1^N) \sum_{s_1^T} p(s_1^T, x_1^T | w_1^N) \\
&= \prod_{n=1}^{N} p(w_n | w_1^{n-1}) \sum_{s_1^T} \prod_{t=1}^{T} p(s_t, x_t | s_1^{t-1}, x_1^{t-1}, w_1^N)
\end{aligned}
$$

- Inverted HMM:

$$
\begin{aligned}
p(w_1^N | x_1^T) &= \sum_{t_1^N} p(w_1^N, t_1^N | x_1^T) \\
&= \sum_{t_1^N} \prod_{n=1}^{N} p(w_n, t_n | w_1^{n-1}, t_1^{n-1}, x_1^T)
\end{aligned}
$$

[Doetsch et al., 2016, Inverted HMM - a Proof of Concept]

Unsolved Problems for End-to-End ASR

- Error rates: Still higher than traditional HMM-based system (one exception).
- Global search: Still a transducer-based or HMM-based search.
- Acoustic model: Word and character-based End-to-End learning.
- Language model: No integration with the language model in training yet.

Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., and Bengio, Y. (2016).
End-to-end attention-based large vocabulary speech recognition.
In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 4945–4949.

Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015).
Attention-based models for speech recognition.
In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 577–585.

Collobert, R., Puhrsch, C., and Synnaeve, G. (2016).
Wav2letter: an end-to-end convnet-based speech recognition system.

CoRR, abs/1609.03193.

Doetsch, P., Hegselmann, S., SchlÃ$\frac{1}{4}$ter, R., and Ney, H. (2016).
Inverted hmm - a proof of concept.
In *Neural Information Processing Systems Workshop*, Barcelona, Spain.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006).
Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks.
In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 369–376, New York, NY, USA. ACM.

Graves, A. and Jaitly, N. (2014).
Towards end-to-end speech recognition with recurrent neural networks.

In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1764–1772.

Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., and Schmidhuber, J. (2009).
A novel connectionist system for unconstrained handwriting recognition.
IEEE Trans. Pattern Anal. Mach. Intell., 31(5):855–868.

Kim, S., Hori, T., and Watanabe, S. (2016).
Joint ctc-attention based end-to-end speech recognition using multi-task learning.
CoRR, abs/1609.06773.

Miao, Y., Gowayyed, M., and Metze, F. (2015).
EESEN: end-to-end speech recognition using deep RNN models and wfst-based decoding.

In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2015, Scottsdale, AZ, USA, December 13-17, 2015*, pages 167–174.

Senior, A. W., Sak, H., de Chaumont Quitry, F., Sainath, T. N., and Rao, K. (2015).
Acoustic modelling with CD-CTC-SMBR LSTM RNNS.
In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2015, Scottsdale, AZ, USA, December 13-17, 2015*, pages 604–609.

Soltau, H., Liao, H., and Sak, H. (2016).
Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition.
CoRR, abs/1610.09975.

Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Laurent, C., Bengio, Y., and Courville, A. C. (2017).

Towards end-to-end speech recognition with deep convolutional neural networks.
CoRR, abs/1701.02720.